The NWTC Design-Codes Suite: An Overview

Marshall L. Buhl Jr.*
National Renewable Energy Laboratory, Golden, Colorado 80401

Over the years, the National Renewable Energy Laboratory's National Wind Technology Center (NWTC) has created a suite of design and analysis software tools for horizontal-axis wind turbines. The suite includes aeroelastic turbine simulators of various complexity, turbulent wind simulators, tools to aid in the design of control systems, pre- and post-processing tools, and scripts to automate the running of simulations. A major goal of the NWTC staff in recent years has been to automate the flow of data between the programs to reduce the time and effort required to generate loads documents for type certification. The interoperability of the tools has been improved, and several codes have received significant upgrades in functionality, performance, and usability. This paper describes how the codes work together, lists recent improvements, and discusses proposed enhancements.

I. INTRODUCTION

Researchers at the National Wind Technology Center (NWTC) of the National Renewable Energy Laboratory (NREL) have developed a comprehensive suite of software tools to aid in the design of horizontal-axis wind turbines (HAWTs) and to process simulation predictions and test data. Some of the programs developed by NWTC engineers many years ago now require updates to aid in maintenance and ease of use. Of the feedback the laboratory has received about the codes, the most common suggestion is that the users would like the codes to work together better. Although some of the programs require similar kinds of information as others in the suite, they sometimes required the information in different formats. When engineers switch tools, converting the information from one format to another wastes valuable time. Not only did the programs require the data in different formats, but also the formats of the input files varied. With these weaknesses in mind, engineers at the NWTC have put much effort into making their codes more compatible. Some of the codes now share their input files with other programs, and those that must be different share a similar look.

Our code suite includes:

- Programs that simulate the aeroelastic response and performance of HAWTs
- Codes that are used to prepare data files needed by the simulators, analyze the results of the simulators, and analyze test data
- Utility programs that automate the execution of series of simulations. This eliminates the tedious and mistake-prone requirement of manually editing input files and renaming output files for all the cases.

Brief descriptions of the various codes follow, and an overview of how the codes interrelate is shown in Figure 1. Detailed descriptions of the various codes are available on the NWTC Design Codes web site.[†1] Program archives on the web site include source code, executables (when appropriate), and sample input and output files. All the programs are in the public domain—anyone may use excerpts of our source code in her or his own programs.

* Sr. Engineer, National Wind Technology Center, 1617 Cole Blvd./3811, AIAA Member.

† All the codes on our web server are freely available to all, but we only have time to support our partners.

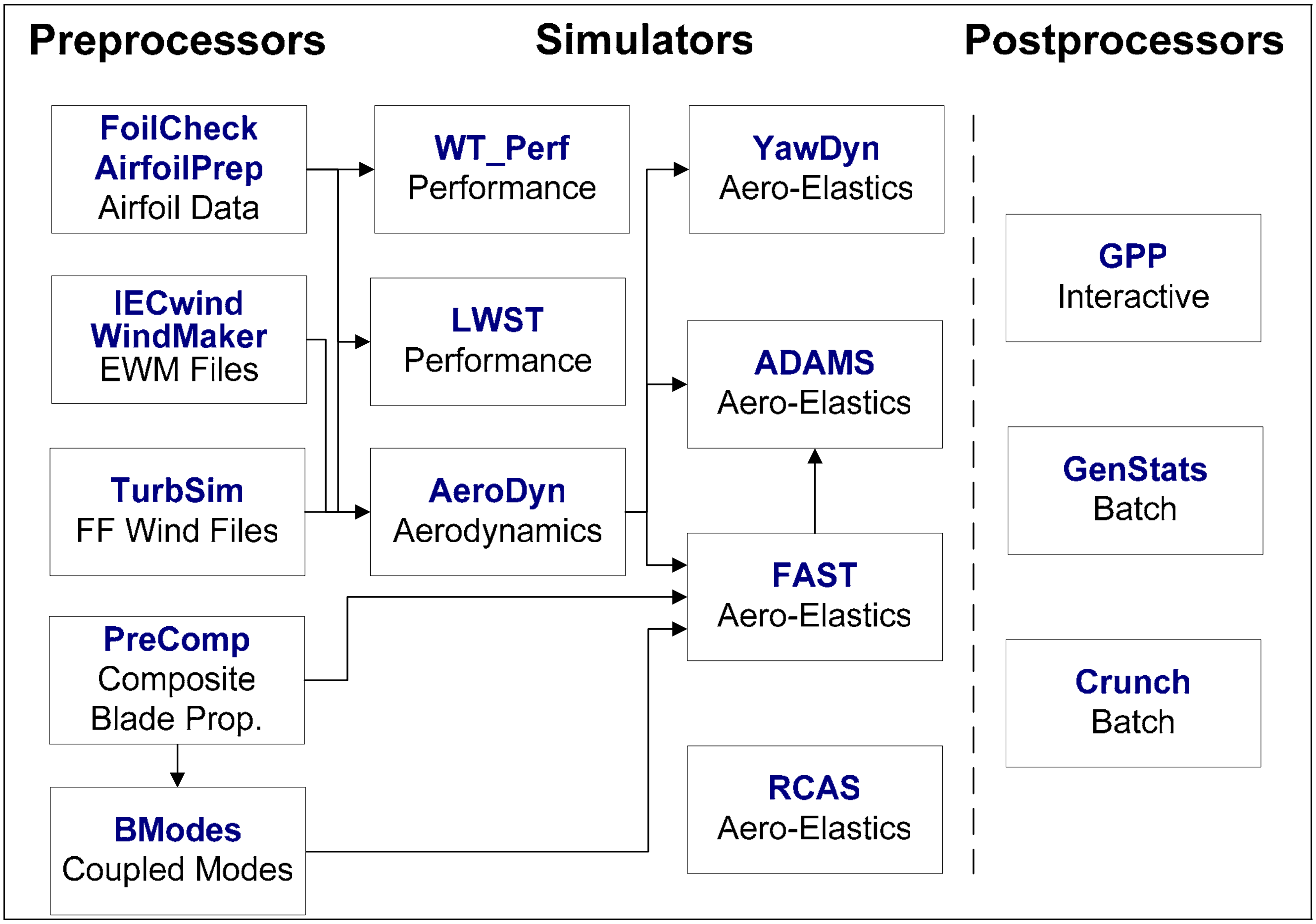

Figure 1. A overview of the design process using the NWTC Code Suite.

II. PREPROCESSORS

A. FoilCheck

We use several programs to create input files for our aeroelastic and performance simulators. One of these preprocessors is FoilCheck,[2] which helps generate airfoil tables. FoilCheck is a simple Fortran program that includes the abilities to expand the airfoil tables via the Viterna[3] method and to compute the dynamic stall parameters needed by AeroDyn.[4-7] AirfoilPrep includes all FoilCheck's features and can modify the airfoil tables for rotational augmentation and interpolate airfoil tables for additional blade stations.

B. AirfoilPrep

Craig Hansen of Windward Engineering recently created a useful spreadsheet called AirfoilPrep,[8] which is a superset of FoilCheck. If you know the airfoil properties for a limited number of blade stations, it can interpolate the aerodynamic coefficients for other span locations. It can also apply the same rotational augmentation corrections for 3-D delayed stall that are used by AeroDyn.

C. IECWind and WindMaker

IECWind[9] and WindMaker[10] generate files that contain the time histories of the International Electrotechnical Commission (IEC) extreme-wind events.[11] IECWind has a simple batch interface; WindMaker has a graphical user interface.

D. SNLWIND-3D, SNwind, and TurbSim

Engineers can use SNLWIND-3D,[12] SNwind,[13,14] and TurbSim[15,16] to generate stochastic, full-field or hub-height turbulence files that can be used by Garrad Hassan's *BLADED*[17,18] and all our AeroDyn-based aeroelastic codes.

Our next-generation wind simulator, TurbSim, is a major rewrite of SNwind and SNLWIND-3D. To drive the tower aerodynamics of AeroDyn, TurbSim has the option to add a line of tower grid points. Other features include coherent wind structures, wind-direction shear, improved algorithms that allow much denser grids, a choice of pseudo-random-number generators, and greater flexibility in grid layout. The streamlined algorithms have dramatically reduced the memory requirements and execution times. Grids are no longer limited to squares of even numbers. They can now be rectangular and may have an odd number of points, which permits a grid point to be centered on the hub. TurbSim also includes spectral models that are valid to much greater heights, as is necessary for today's gigantic wind turbines.

In 2005, we intend to analyze test data from our long-term atmospheric tests in Lamar, Colorado to develop new custom spectra for TurbSim that are suitable for use in the Great Plains. We also hope to use data from the tall towers there to refine spectra for higher elevations. Another planned improvement is to update TurbSim to conform to the upcoming third edition of the IEC standards[11] for wind-turbine design.

E. BModes

We are working on BModes, a new finite element structural code that uses distributed blade mass and stiffness properties to generate mode shapes that couple the flap, lag, and torsional modes. The properties can come from PreComp, test data, or other sources. The mode shapes are compatible with FAST.[19,20]

F. PreComp

Another new program under development, PreComp, uses the external shape of blades, material properties, and composite lay-up to generate the distributed blade mass and stiffness properties, which can be used by various aeroelastic codes and BModes.

III. SIMULATORS

We have some codes that simulate the performance of rotors and others that simulate the aeroelastic response of entire wind turbines.

A. WT_Perf

WT_Perf[21,22] uses blade-element/momentum (BEM) theory to model the steady, rotor performance of HAWTs. It is simple to use and extremely fast. It has the option to apply tip or hub losses via Prandtl's method[23]. The user can enable or disable the effects of tangential induction. The drag terms may be included in the induction equations. Users can choose either the classical brake-state induction calculation or the advanced brake-state calculation, which allows use of Gauert's empirical relationship for high induction factors. For the advanced brake state, the user can choose the traditional PROP-PC algorithm or the PROPX[24] algorithm, which is similar to the AeroDyn algorithm. The two algorithms give slightly different results. Although we are uncertain of its accuracy, there is an option to enable skewed-wake corrections for off-axis flow. The algorithm came from AeroDyn. WT_Perf can generate performance data for all combinations of parametrically varied wind speed (or tip-speed ratio), blade pitch, and rotor speed. WT_Perf's airfoil tables are the same as those used by AeroDyn.

We completely rewrote and modernized WT_Perf in FY2004. To simplify support and maintenance, we moved routines commonly used by WT_Perf and many of our other codes to a separate library called the NWTC Subroutine Library.[25] We enhanced the dataflow between it and AeroDyn to reduce the time required to build models and lessen the chances of errors. Choices of induction algorithms, skewed-wake corrections, variable segment lengths, Reynolds Number (Re) variation, and new ways of specifying turbine states have improved the accuracy and usability of the simulator.

In FY2005, we would like to add a GDW algorithm to WT_Perf. One problem with BEM theory is that it is not valid for off-axis flow. Although we have added a skewed-wake correction to the BEM algorithm in WT_Perf, we question its accuracy. We need to be able to accurately predict the power of turbines with off-axis flow because turbines often operate with a yaw error, manufacturers commonly use shaft tilt to improve tower clearance in HAWTs, and sloped terrain can produce a vertical wind component. The current version of WT_Perf has inputs for yaw error and shaft tilt, but not for vertical wind components. We will probably add capability for the latter once we have a trustworthy skewed-wake correction.

Engineers often find that their HAWT performance predictions disagree with test data, so they tune the airfoil characteristics to the data. This tuning can be time-consuming and tedious. Beyond FY2005, we may add an automated tuning capability to WT_Perf. Such an approach can be fraught with danger in that tuning properties to minimize the difference from test data for one parameter or set of parameters can increase the differences for other parameters. Creating a useful weighted cost function may be more art than science.

If we add an optimization algorithm to WT_Perf, we may also use it to make initial designs for optimized blades. Once again, defining *optimal* may be the most difficult part of the task. Optimizing a rotor for maximum performance does not necessarily create a turbine with the lowest cost of energy. Instead of doing a purely aerodynamic optimization, we will most likely have to add structural considerations to the optimization. It may also be difficult to find realistic constraints for the optimization scheme. Without reasonable constraints, an optimizer may produce truly bizarre blades.

We may also add the lifting-surface/prescribed-wake algorithm used in Kocurek's Lifting Surface Wind Turbine (LSWT) program to WT_Perf because WT_Perf is more compatible than LSWT with the rest of our code suite. However, the effort may be too great. We will have to investigate it further before we can decide if it is worth the effort.

We may add new models for tip losses. Some researchers use the Goldstein method,[26] and AeroDyn now has the option to use the tip-loss model developed by Georgia Institute of Technology.[27] However, that model appears to apply only to the turbine used in Phase VI of NREL's Unsteady Aerodynamics Experiment (UAE).[28]

B. LSWT

David Kocurek of Computational Methodology Associates developed LSWT[29] for the NWTC during the 1980s. This prescribed-wake code can model the trailing vorticity distribution of the wake. This model includes tip, root, and midspan trailing vorticity. LSWT is understandably more difficult to use than WT_Perf and execution times are much longer, but it does a superior job of predicting angle-of-attack distributions in the stall and post-stall regions.[30]

We will continue to compare LSWT predictions to data from the UAE Ames wind-tunnel test.[31] We hope to use these studies to gain a better understanding of stall physics. We would like to add the capability to model tip winglets. We have also noticed some numerical problems associated with the number of spanwise and chordwise panels used in the analysis.

C. AeroDyn

AeroDyn[4-7] is a library of subroutines that can be linked with a structural code to provide the lift, drag, and pitching-moment forces at the blade or tower nodes. Windward Engineering developed most of the routines under contract to the NWTC. AeroDyn currently works with YawDyn,[32,33] FAST, and ADAMS.[*,34,35] To calculate the induced velocities, AeroDyn has two options: the PROPX-style BEM theory and a generalized dynamic wake (GDW) that uses Peters and HaQuang's method. The BEM induction can optionally include tip and hub losses based on Prandtl's work as described by Glauert.[36] AeroDyn also has an option to use dynamic stall based on the work of

[*]ADAMS is a commercial program produced by MSC.Software.

Leishman and Beddoes.[37-39] AeroDyn can excite the rotor structure with steady or turbulent, sheared, hub-height winds, or by using full-field stochastic winds such as those generated by TurbSim. SNwind and TurbSim can also generate stochastic hub-height winds for AeroDyn. Users can also write their own wind routines to read special-format wind files or to create analytical winds. For example, we used this feature to model a Rankine vortex with varied parameters such as radius and circulation strength.[40]

In FY2004, Windward Engineering and engineers at the NWTC modified AeroDyn to compute tower aerodynamics. Windward also added an upwind tower wake and automatic switching between the GDW algorithm and the BEM algorithm for low wind speeds. We have also added rotational augmentation corrections for 3-D delayed stall. The corrections are for both lift and drag. It uses the Du method[41] to augment the lift and the Eggers method[42] for the drag. Another significant change is the ability to add coherent wind structures on top of normal background turbulence. We also recently produced a theory manual for AeroDyn.

In FY2005, we plan to significantly rewrite AeroDyn. In addition to modifying it to use the NWTC Subroutine Library, we hope to simplify the interface to the various structural codes that call AeroDyn. We are enlisting help from Davis Peters at the University of Washington to help us replace the linear GDW model with the newer, nonlinear model.[43] This will include at least 33 flow states and Cheng Jian He's correction for the vortex ring state. This should eliminate the instability problems the old GDW model had at low wind speeds. We will also add the new format of airfoil tables that we developed for WT_Perf. This new format removes the restriction that all the tables for different Re values use the same angle-of-attack list. We will also make minor changes to the main AeroDyn input file so that inputting option flags and strings is similar to the rest of our code suite.

Other plans are to continue our comparisons of AeroDyn to data from the Ames 80 x 120 wind-tunnel test of the UAE turbine.[44] We may also do some comparisons to computational fluid dynamics (CFD) predictions.

Longer term, we may add vortex-filament/free-wake algorithms and prescribed-wake algorithms to AeroDyn. We will also continue our comparisons to the UAE wind-tunnel test data and to CFD predictions.

D. YawDyn

YawDyn,[32,33] also developed by the engineers at Windward Engineering, is our simplest HAWT-specific aeroelastic simulator. It can model two- or three-bladed turbines with up to four degrees of freedom (DOFs). Two-bladed turbines can have yaw and teeter DOFs. Three-bladed turbines can have a yaw DOF and one flap DOF for each blade. YawDyn is typically an order of magnitude faster than real time, and easier to use than our other aeroelastic simulators. It uses the AeroDyn subroutine library for its aerodynamic calculations.

E. FAST

Our workhorse aeroelastic simulator, called FAST,[19,20] usually runs several times faster than real time on current PCs. This medium-complexity, HAWT-specific, turbine simulator uses up to 18 DOFs, and includes flexible blades and a flexible tower. FAST uses modal representation to model the flexibility. It has a linearization feature that generates state-space matrices that one can use to design control systems or to perform full-system model analysis. It can even be called as a plant model from Simulink (an add-in for The Mathworks' MATLAB*). FAST can be used as a preprocessor to generate input files to model HAWTs with the ADAMS general-purpose, multibody-dynamics code.

In FY2004, we enhanced FAST to include furling DOFs and tail fin aerodynamics that enable the modeling of most small wind turbines. The furling DOFs take the form of hinges between the tower and rotor and between the tower and tail. Users can specify the location and orientation of the hinges. The FAST-to-ADAMS preprocessor, linearization, and the interface to Bladed-style DLL master controllers are also fairly new features. We recently interfaced FAST to Simulink for streamlined controls design and improved the interface between FAST and typical controls routines. Included in that is the ability to control the yaw position. To help in the understanding of wind-turbine noise, FAST now includes semi-empirical aeroacoustic calculations.[45]

We plan to add tower and nacelle aerodynamics to FAST in FY2005. We are also adding a six-DOF foundation that can model standard foundations, earthquake loading, and offshore foundations with wave loading. Instead of

---

*MATLAB is a commercial program developed by The Mathworks.

two flap DOFs and one edge DOF, FAST will have four coupled DOFs per blade. Users can select any combination of flap, edge, or torsional DOFs. We also plan to add a tower-torsion DOF.

We are doing some experimental work with noise predictions, and hope to upgrade the noise prediction capabilities now in FAST.[46] We are investigating a new algorithm for predicting noise caused by turbulent inflow. We are also using XFOIL[47] to study boundary-layer thickness predictions and we are studying turbulence length scales.

We have big plans for long-term FAST development. We plan to add two shaft-bending DOFs, guy wires, and the ability to specify counterclockwise rotation in addition to the current clockwise-only rotation. Although FAST can currently generate ADAMS models for turbines with blade precurve and presweep, we plan to add the ability to model such blades directly in FAST. To make FAST more user friendly, we hope to be able to generate animations of simulations and mode shapes. We plan to add multi-blade coordinate transformations to eliminate the periodicity in the linearization for three-bladed turbines.

Small wind turbines have such a wide variation in rotational speed that we need to be able to adjust the integration time step so that we do not waste time at slow rotor speeds, but still have accurate results at high speeds. We will investigate available variable-step integration schemes to solve this problem.

Currently, we require users to use an external program to generate mode shapes for the blades and tower. We hope to include that code within FAST in the future to eliminate that extra step. It will also allow FAST to recompute the mode shapes whenever the blade pitch or rotor speed changes significantly.

F. ADAMS2AD

ADAMS2AD[5,48] is a library of routines that allow ADAMS[‡,35] to call AeroDyn routines when modeling HAWTs. When using ADAMS with AeroDyn, users must compile ADAMS2AD and AeroDyn to a dynamic-link library, which is then called from ADAMS. Being a general-purpose code with virtually unlimited DOFs (it can also model robots and cars), ADAMS is significantly slower than FAST, but is more versatile. FAST is limited to most standard types of HAWTs, but ADAMS with AeroDyn can model virtually any kind of HAWT.

G. RCAS

Although we do not distribute it on our web site, we also use RCAS (Rotorcraft Comprehensive Analysis System), which comes from the U.S. Army's Aeroflightdynamics Directorate (AFDD). AFDD and Advanced Rotorcraft Technology (ART) developed RCAS to model rotorcraft, but we have tested it thoroughly[49-51] and found it to be a useful tool for modeling wind turbines. RCAS is a general-purpose code much like ADAMS, but it uses finite elements instead of a lumped-mass approach to model flexible structures. Unlike ADAMS, it has its own rotor aerodynamics models and users have a wide selection of algorithms from simple BEM to CFD. Like FAST and unlike ADAMS, it has built-in capability to linearize rotating structures to create state-space matrices. It also has built-in stability analysis capability. We are currently testing this feature.

As larger turbines are built, there is some concern that they may become aeroelastically unstable. In FY2005, we plan to develop an RCAS model of a multi-megawatt wind turbine with curved blades to establish the stability boundaries.

Longer term, we plan to continue our studies of aeroelastic stability by exploring its boundaries with RCAS. We also plan to identify turbine-specific scaling parameters that influence diverse instabilities. We also hope to understand the couplings underlying instabilities and adverse dynamic loads. Afterward, we plan to create a list of suggested design guidelines for aeroelastic stability.

H. Advanced Codes

The Sandia National Laboratories and NREL have subcontracts with researchers in private industry and academia for new codes that use advanced aerodynamic models. Gordon Leishman is working on a free-vortex wake method at the University of Maryland.[52,53] For the very long term, several groups are also studying CFD.

IV. POSTPROCESSORS

A. Crunch and CombEEv

Our most popular postprocessing tool, Crunch,[54,55] is a batch-style analysis program. An input file specifies which of many types of analyses to do and a list of files to analyze. Users can process multiple files individually or as a single aggregated file. Table 1 lists the primary capabilities of Crunch.

Because the current version of Crunch cannot handle multiple files with different lengths, separate Crunch runs must be used for groups of files of the same length. This causes a problem when a user wants to tabulate the extreme events for all the test or simulation data. For instance, discrete IEC extreme-wind events are only a few seconds long, but IEC normal-turbulence-model (NTM) runs usually last for 10 minutes. One Crunch run can process all the discrete events and another can process all the long turbulence simulations. We wrote a Perl script called CombEEv,[56,57] which will find the global extremes from multiple extreme-event tables and generate a single table.

In FY2005, we intend to improve the user interface of Crunch to add more information to output files and new checks for data validity. We intend to augment the rainflow cycle counting to generate damage-equivalent loads. Users have asked us to add geometric correction factors to the load-rose calculations and partial safety factors to extreme loads. To enhance the power of the calculated channels, we will add a logical IF function similar to those used in spreadsheets. Because some calculations require a time column and because not all data files have one, we will add a feature that engineers can use to generate one.

Table 1. Crunch features.

- Scales and offsets
- Peak fitting
- Low-, high-, band-pass IIR filtering
- Calculated channels
- Moving averages
- Load roses
- Azimuth averages
- Crosstalk removal
- Peak/trough listing
- Probability mass
- Rainflow cycle counting
- Extreme-event tabulation
- Extreme-value extrapolation
- Statistics
- Summary files

B. GPP

GPP[58,59] is our oldest postprocessor. Unlike Crunch, it is an interactive program. That makes it useful for reading in a data set and manipulating or analyzing the data in many ways, but less useful for processing a large number of files. GPP and Crunch have many common features, but each program has capabilities the other does not. We are gradually adding GPP features to Crunch, but not the reverse. GPP is essentially static and its only changes now are bug fixes. Table 2 lists the primary capabilities of GPP.

Table 2. GPP features.

- Azimuth averaging
- Binning
- Interpolation
- LSQ fit
- Probability mass
- PSDs
- Rainflow cycle counting
- Statistics
- Filtering
- Data-record limiting
- Row trimming
- Column trimming
- Multi-file merging
- Calculated channels
- Units conversion
- Generate time columns

C. GenStats

Crunch started out as a much simpler program called GenStats.[60,61] As its name would suggest, GenStats only generates statistics. We still use it because it does not require all files to be the same length, as Crunch does. We hope to remove that restriction in Crunch, so we may eventually remove GenStats from our code suite.

V. UTILITIES

A. RunIEC, RunNTM, and CondorNTM

We wrote some Perl scripts to automate running large numbers of simulations. The first is called RunIEC,[62,63] which will run FAST for each IEC extreme-wind event. The output files are each given names that reflect the cases being run. This automation of a tedious task reduces the chance of errors.

Similarly, RunNTM[64,65] runs a series of TurbSim and FAST simulations to process a large number of IEC NTM cases. A user can specify a series of wind speeds, turbulence categories, a starting random seed, and a seed increment, and RunNTM will run all the cases. Running a series of 20 wind speeds with 10 random cases each would be a daunting task without a script such as RunNTM to automate it. Output file names include the input parameters that defined the runs.

Running many turbulence simulations can take a lot of time on a single PC. We once had a research project where we needed to simulate an entire year of turbine operation. This project would not have been possible if we had had to resort to serial processing on one (or even a few) PCs. We leveraged the NWTC's investment in computers by enlisting all our newer computers for the task. We used a freeware program called Condor[66] from the University of Wisconsin. We modified the RunNTM script to submit simulation jobs to our pool of computers, which ran the simulations during their idle time. With a couple dozen processors running nearly around the clock, a project that would have taken more than a year on a single PC took only a few weeks. We offer on our web server an example script called CondorNTM[67] and a set of files to work with it that will submit NTM simulations to a Condor pool to run them in parallel instead of sequentially.

B. NWTC Subroutine Library

To reduce maintenance and development costs and leverage the investment in making input files more compatible, the NWTC has developed a library of standard routines, the NWTC Subroutine Library. Linking different programs with this library obviates the need to reproduce the logic that is common to many of our codes.

Historically, many of our programs used collections of routines that were of general use, but each had its own copy and subset. Occasionally, we would find better ways to perform some of these tasks and would have to update all the programs that used those routines. By keeping a single set of routines in a central location, we have reduced our workload.

The files in the library include some routines that are compiler independent and some that we designed to work with a specific compiler. All compiler-specific logic and global data are stored in separate files. This makes it relatively easy to port our programs from one compiler or operating system to another. We have striven to eliminate all system-specific logic and data from the file collections of the individual codes to reduce the effort to port them. As of this writing, we include only logic and global variables in the library for the Visual Fortran family of Fortran compilers for the Windows operating system.

ACKNOWLEDGMENTS

I would like to thank Craig Hansen and David Laino of Windward Engineering for their work in keeping AeroDyn current, creating AirfoilPrep, and comparing simulation predictions to the data collected in the Unsteady Aerodynamics Experiment in the NASA/Ames wind tunnel. For help in describing program features, setting goals, tracking down references, and for attempting to educate me, I would like to thank my coworkers: Gunjit Bir, Maureen Hand, Bonnie and Jason Jonkman, Neil Kelley, Pat Moriarty, and Jim Tangler. I would also like to thank Kelly Jackson, Stefanie Woodward, Ruth Baranowski, and Kathy O'Dell for polishing my prose.

REFERENCES

[1]National Renewable Energy Lab., *NWTC Design Codes Web Page*, http://wind.nrel.gov/designcodes/, National Renewable Energy Lab., NREL/EL-500-28553, Golden, CO, 22 Sept. 2004.

[2]National Renewable Energy Lab., *NWTC Design Codes (FoilCheck)*, http://wind.nrel.gov/designcodes/foilcheck/, National Renewable Energy Lab., Golden, CO, 21 Nov. 2002.

[3]Viterna, L. A., Corrigan, R. D., "Fixed Pitch Rotor Performance of Large Horizontal Axis Wind Turbines," *Presented at the DOE/NASA Workshop on Large Horizontal Axis Wind Turbines*, 28–30 July 1981, Cleveland, OH.

[4]National Renewable Energy Lab., *NWTC Design Codes (AeroDyn)*, http://wind.nrel.gov/designcodes/simulators/aerodyn/, National Renewable Energy Lab., Golden, CO, 3 Oct. 2003.

[5]Laino, David J., Hansen, A. Craig, “User's Guide to the Wind Turbine Dynamics Aerodynamics Computer Software AeroDyn,” prepared for the National Renewable Energy Lab. under Subcontract No. TCX-9-29209-01, Windward Engineering, Salt Lake City, UT, Dec. 2002, 57 pp.

[6]Laino, David J., Hansen, A. Craig, “User's Guide to the Computer Software Routines AeroDyn Interface for ADAMS®,” prepared for the National Renewable Energy Lab. under Subcontract No. TCX-9-29209-01, Windward Engineering, Salt Lake City, UT, Sept. 2001, 24 pp.

[7]Moriarty, Patrick J., Hansen, A. Craig, “AeroDyn Theory Manual,” National Renewable Energy Lab., NREL/EL-500-36881, Golden, CO, Oct. 2004.

[8]National Renewable Energy Lab., *NWTC Design Codes (AirfoilPrep)*, http://wind.nrel.gov/designcodes/preprocessors/airfoilprep/, National Renewable Energy Lab., Golden, CO, 27 Aug. 2004.

[9]National Renewable Energy Lab., *NWTC Design Codes (IECWind)*, http://wind.nrel.gov/designcodes/preprocessors/iecwind/, National Renewable Energy Lab., Golden, CO, 21 Nov. 2002.

[10]National Renewable Energy Lab., *NWTC Design Codes (WindMaker)*, http://wind.nrel.gov/designcodes/preprocessors/windmaker/, National Renewable Energy Lab., Golden, CO, 21 Nov. 2002.

[11]International Electrotechnical Commission (TC88), *Wind Turbine Generator Systems—Part 1: Safety Requirements*, Second Edition, International Electrotechnical Commission, IEC 61400-1, Geneva, Switzerland, Feb. 1999.

[12]National Renewable Energy Lab., *NWTC Design Codes (SNLWIND-3D)*, http://wind.nrel.gov/designcodes/simulators/snlwind3d/, National Renewable Energy Lab., Golden, CO, 21 Nov. 2002.

[13]National Renewable Energy Lab., *NWTC Design Codes (SNwind)*, http://wind.nrel.gov/designcodes/simulators/snwind/, National Renewable Energy Lab., Golden, CO, 19 Nov. 2003.

[14]Buhl, M. L., Jr., “SNwind User's Guide,” National Renewable Energy Laboratory, NREL/EL-500-30121, Golden, CO, June 2003, 4 pp.

[15]National Renewable Energy Lab., *NWTC Design Codes (TurbSim)*, http://wind.nrel.gov/designcodes/simulators/turbsim/, National Renewable Energy Lab., Golden, CO, 2004.

[16]Jonkman, Bonnie J., Buhl, M. L., Jr., “TurbSim User's Guide,” National Renewable Energy Lab., NREL/EL-500-36970, Golden, CO, 2004, 7 pp.

[17]Bossanyi, E. A., “GH Bladed Theory Manual,” Garrad Hassan and Partners Limited, Document No. 282/BR/009, Bristol, England, Dec. 2003.

[18]Bossanyi, E. A., “GH Bladed Version 3.6 User Manual,” Garrad Hassan and Partners Limited, Document No. 282/BR/010, Bristol, England Dec. 2003.

[19]National Renewable Energy Lab., *NWTC Design Codes (FAST)*, http://wind.nrel.gov/designcodes/simulators/fast/, National Renewable Energy Lab., Golden, CO, 7 Apr. 2004.

[20]Jonkman, J. M., Buhl, M. L., Jr., “FAST User's Guide,” National Renewable Energy Lab., NREL/EL-500-29798, Golden, CO, Mar. 2004, 105 pp.

[21]National Renewable Energy Lab., *NWTC Design Codes (WT_Perf)*, http://wind.nrel.gov/designcodes/simulators/wtperf/, National Renewable Energy Lab., Golden, CO, 2 Sept. 2004.

[22]Buhl, M. L., Jr., “WT_Perf User's Guide,” National Renewable Energy Lab., NREL/EL-500-29382, Golden, CO, 31 Aug. 2004, 4 pp.

[23]Hibbs, B., Radkey, R. L., “Small Wind Energy Conversion Systems (SWECS) Rotor Performance Model Comparison Study,” Prepared for Rockwell International Corporation under Subcontract No. PFN-13470W, Aerovironment, Inc., Pasadena, CA, Nov. 1981, 63 pp.

[24]Harman, C. Ross, “PROPX: Definitions, Derivations, and Data Flow,” Oregon State University, Corvalis, OR, Aug. 1994, 57 pp.

[25]National Renewable Energy Lab., *NWTC Design Codes (NWTC Subroutine Library)*, http://wind.nrel.gov/designcodes/miscellaneous/nwtc_subs, National Renewable Energy Lab., Golden, CO, 14 Sept. 2004.

[26]Goldstein, S., “On the Vortex Theory of Screw Propeller,” *Proc. Roy. Soc. (A)*, Vol. 123, No. 440, 1929.

[27]Xu, Guanpeng, Sankar, Lakshmi N., “Application of a Viscous Flow Methodology to the NREL Phase VI Rotor,” *Collection of the 2002 ASME Wind Energy Symposium Technical Papers Presented at the 40th AIAA Aerospace Sciences Meeting and Exhibit,* AIAA, Washington, DC, and ASME, New York, Jan. 2002, pp. 83–93.

[28]Hand, M. M., Simms, D. A., Fingersh, L. J., Jager, D. W., Cotrell, J. R., Schreck, S., Larwood, S. M., “Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns,” National Renewable Energy Lab., NREL/TP-500-29955, Golden, CO, Dec. 2001, 298 pp.

[29]Kocurek, D., “Lifting Surface Performance Analysis for Horizontal Axis Wind Turbines,” Solar Energy Research Inst., SERI/STR-217-3163, Golden, CO, 1987, 316 pp.

[30]Tangler, J. L., “Insight into Wind Turbine Stall and Post-Stall Aerodynamics,” *Wind Energy*, Vol. 7, No. 3, 2004, pp. 247–260.

[31]Tangler, J., “Nebulous Art of Using Wind-Tunnel Airfoil Data for Predicting Rotor Performance,” *Collection of the ASME Wind Energy Symposium, Presented at the 40th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Washington, DC, and ASME, New York, Jan. 2002, pp. 190–196.

[32]*NWTC Design Codes (YawDyn)*, http://wind.nrel.gov/designcodes/yawdyn/, National Renewable Energy Lab., Golden, CO, 25 Apr. 2003.

[33]Laino, David J., Hansen, A.C., “User's Guide to the Wind Turbine Dynamics Computer Program YawDyn,” Prepared for the National Renewable Energy Laboratory under Subcontract No. TCX-9-29209-01, Windward Engineering, Salt Lake City, UT, Jan. 2003, 30 pp.

[34]Elliott, A. S., McConville, J. B., “Application of a General-Purpose Mechanical Systems Analysis Code to Rotorcraft Dynamics Problems,” *Prepared for the National Specialists' Meeting on Rotorcraft Dynamics*, American Helicopter Society, Alexandria, VA, 1989.

[35]Elliott, A. S., “Analyzing Rotor Dynamics with a General-Purpose Code,” *Mechanical Engineering*, Vol. 112, No. 12, ASME, New York, 1990, pp. 21–25.

[36]Glauert, H., “Airplane Propellers,” *Aerodynamic Theory*, edited by W. F. Durand, Berlin, Julius Springer, 1935.

[37]Leishman, J. G., and Beddoes, T. S., “A Generalized Model for Airfoil Unsteady Behavior and Dynamic Stall Using the Indicial Method,” *Proceedings from the 42nd Annual Forum of the American Helicopter Society*, Washington, DC, 1986, pp. 243–266.

[38]Leishman, J. G., and Beddoes, T. S., “A Semi-Empirical Model for Dynamic Stall,” *Journal of the American Helicopter Society*, Vol. 34, No. 3, 1989, pp. 3-17.

[39]Leishman, J. G., “Modeling Sweep Effects on Dynamic Stall,” *Journal of the American Helicopter Society*, Vol. 34, No. 3, 1989, pp. 18-29.

[40]Hand, M. M., “Mitigation of Wind Turbine/Vortex Interaction Using Disturbance Accommodating Control,” National Renewable Energy Lab., NREL/TP-500-35172, Golden, CO, Dec. 2003, 125 pp.

[41]Du, Zhaohui, Selig, Michael S., “A 3-D Stall-Delay Model for Horizontal Axis Wind Turbine Performance Prediction,” *Collection of the 1998 ASME Wind Energy Symposium Technical Papers Presented at the 36th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Washington, DC, ASME, New York, 1998, pp. 9–19.

[42]Eggers, A. J., Jr., Chaney, K., Digumarthi, R., “An Assessment of Approximate Modeling of Aerodynamic Loads on the UAE Rotor,” *Collection of the 2003 ASME Wind Energy Symposium Technical Papers Presented at the 41st AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Washington, DC, ASME, New York, 2003, pp. 283–292.

[43]Su, Ay, Yoo, Kyung M., Peters, David A, “Extension and Validation of an Unsteady Wake Model for Rotors,” J. of Aircraft, Vol. 29, No. 3, 1992, pp. 374–383.

[44]Laino, David J., Hansen, A. Craig (), “Current efforts toward Improved Aerodynamic Modeling Using the AeroDyn Subroutines,” *Collection of the 2004 ASME Wind Energy Symposium Technical Papers Presented at the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Washington, DC, ASME, New York, 2004, pp. 329–338.

[45]Moriarty, P., Migliore, P., “Semi-Empirical Aeroacoustic Noise Prediction Code for Wind Turbines,” National Renewable Energy Lab., NREL/TP-500-34478, Golden, CO, 2003, 39 pp.

[46]Moriarty, P. J, (2004), “Development and Validation of a Semi-Empirical Wind Turbine Aeroacoustic Code,” *Collection of the 2004 ASME Wind Energy Symposium Technical Papers at the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Washington, DC, ASME, New York, 2004, pp. 577–586.

[47]XFOIL, Subsonic Airfoil Development System, Software Package, Ver. 6.94, Massachusetts Institute of Technology, Cambridge, MA, Dec. 2001.

[48]National Renewable Energy Lab., *NWTC Design Codes (ADAMS2AD)*, http://wind.nrel.gov/designcodes/adams2ad/, National Renewable Energy Lab., Golden, CO, 15 Mar. 2004.

[49]Jonkman, J. M., Cotrell, J., “A Demonstration of the Ability of RCAS to Model Wind Turbines,” National Renewable Energy Lab., NREL/TP-500-34632, Golden, CO, Aug. 2003, 52 pp.

[50]Bir, G. S., “Structural Dynamics Validation of RCAS (Rotorcraft Comprehensive Analysis System),” National Renewable Energy Lab., NREL/TP-500-35328, Golden, CO, 2004.

[51]Tangler, J., Bir, G., “Evaluation of RCAS Inflow Models for Wind Turbine Analysis, National Renewable Energy Lab., NREL/TP-500-35109, Golden, CO, Feb. 2004, 42 pp.

[52]Gupta, Sandeep, Leishman, J. Gordon, “Stability of Methods in the Free-Vortex Wake Analysis of Wind Turbines,” *Collection of the 2004 ASME Wind Energy Symposium Technical Papers Presented at the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Washington, DC, ASME, New York, 2004, pp. 339–353.

[53]Gupta, Sandeep, Leishman, J. Gordon, “Accuracy of the Induced Velocity of Wind Turbine Wakes Using Vortex Segmentation,” *Collection of the 2004 ASME Wind Energy Symposium Technical Papers Presented at the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Washington, DC, ASME, New York, 2004, pp. 354–367.

[54]National Renewable Energy Lab., *NWTC Design Codes (Crunch)*, http://wind.nrel.gov/designcodes/postprocessors/crunch/, National Renewable Energy Lab., Golden, CO, 2 Jul. 2004.

[55]Buhl, M.L., Jr., “Crunch User's Guide,” National Renewable Energy Lab., NREL/EL-500-30122, Golden, CO, 15 Oct. 2003, 12 pp.
[56]National Renewable Energy Lab., *NWTC Design Codes (CombEEv)*, http://wind.nrel.gov/designcodes/postprocessors/combeev/, National Renewable Energy Lab., Golden, CO, 21 Jun. 2004.
[57]Buhl, M. L., Jr., “CombEEv User's Guide,” National Renewable Energy Lab., NREL/EL-500-31664, Golden, CO, 21 Jun. 2004, 2 pp.
[58]National Renewable Energy Lab., *NWTC Design Codes (GPP)*, http://wind.nrel.gov/designcodes/postprocessors/gpp/, National Renewable Energy Lab., Golden, CO, 17 Sept. 2004.
[59]Buhl, M. L., Jr., Kelley, N. D., Simms, D. A., (1995), “GPP—A General-Purpose Post Processor for Wind Turbine Data Analysis,” *Wind Energy 1995: Proceedings of the Energy and Environmental Expo '95, the Energy-Sources Technology Conference and Exhibition,* SED–Vol. 16, ASME, New York, 1995, pp. 237–242.
[60]National Renewable Energy Lab., *NWTC Design Codes (GenStats)*, http://wind.nrel.gov/designcodes/postprocessors/genstats/, National Renewable Energy Lab., Golden, CO, 21 Nov. 2004.
[61]Buhl, M. L., Jr., *GenStats User's Guide,* National Renewable Energy Lab., NREL/EL-500-31663, Golden, CO, Apr. 2002, 3 pp.
[62]National Renewable Energy Lab., *NWTC Design Codes (RunIEC)*, http://wind.nrel.gov/designcodes/miscellaneous/runiec/, National Renewable Energy Lab., Golden, CO, 21 Nov. 2002.
[63]Buhl, M. L., Jr, (), *RunIEC User's Guide*, National Renewable Energy Lab., NREL/EL-500-29385, Golden, CO, Jun. 2001, 3 pp.
[64]National Renewable Energy Lab., *NWTC Design Codes (RunNTM)*, http://wind.nrel.gov/designcodes/miscellaneous/runntm/, National Renewable Energy Lab., Golden, CO, 21 Nov. 2002.
[65]Buhl, M. L., Jr., *RunNTM User's Guide*, National Renewable Energy Lab., NREL/EL-500-29386, Golden, CO, Jun. 2001, 3 pp.
[66]Condor, Software Package, Ver. 6.6.5, University of Wisconsin, Madison, WI, http://www.cs.wisc.edu/condor/, July 2004.
[67]National Renewable Energy Lab., *NWTC Design Codes (CondorNTM)*, http://wind.nrel.gov/designcodes/utilities/condorntm/, National Renewable Energy Lab., Golden, CO, 2004.